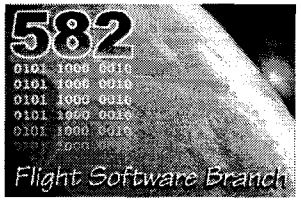


Fight Software Workshop 2007 (FSW-07)

Framework Based Guidance Navigation and Control Flight Software Development

David McComas
Code 582
david.c.mccomas@nasa.gov
301-286-9038

November 5-6, 2007

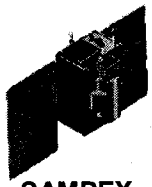
Outline

- **NASA/Goddard Guidance Navigation and Control (GN&C) Flight Software (FSW) Development Background**
- **GN&C FSW Development Improvement Concepts**
- **GN&C FSW Application Framework**

Why the Need for Change ?

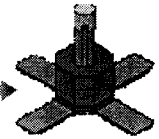
- **The cost of recent GN&C FSW efforts has remained fairly constant while only providing small, if any, incremental functional advancements.**
  - Indicates the FSW development process is not capitalizing on work from previous missions (e.g. no code, test, and document reuse)
  - The GN&C FSW quality remains high, but at a high cost
- **Specific Problem areas**
  - Lack of integrated GN&C analysis and FSW development processes
  - Insufficient GN&C FSW infrastructure, documentation, and unit tests
- **Consequences**
  - Rewrite requirements for each new mission
  - Difficult analyst assessment of existing FSW for new missions
  - Limited integration of GN&C FSW with analyst's simulations
  - Lack of a configuration management (CM) system for reusing assets

GSFC FSW Evolution

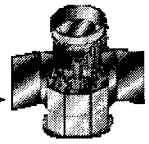


SAMPEX
(launched 8/92)

SWAS
(launched 12/98)

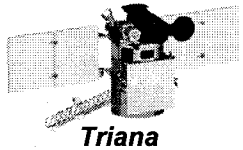


TRACE
(launched 3/98)



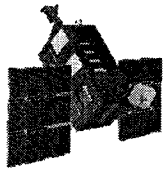
WIRE
(launched 2/99)

SMEX-Lite

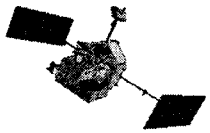


Triana
(waiting for launch)

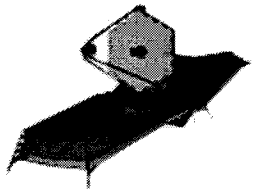
Swift BAT
(12/04)



XTE *(launched 12/95)*

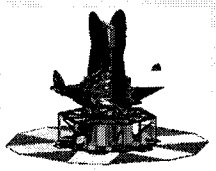


TRMM *(launched 11/97)*

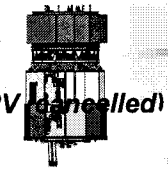


JWST ISIM
(2011)

IceSat GLAS *(01/03)*

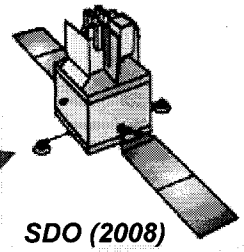


MAP *(launched 06/01)*



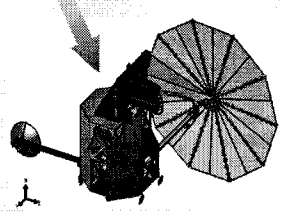
HRV (cancelled)

GPM(2013)



SDO (2008)

ST-5 *(5/06)*

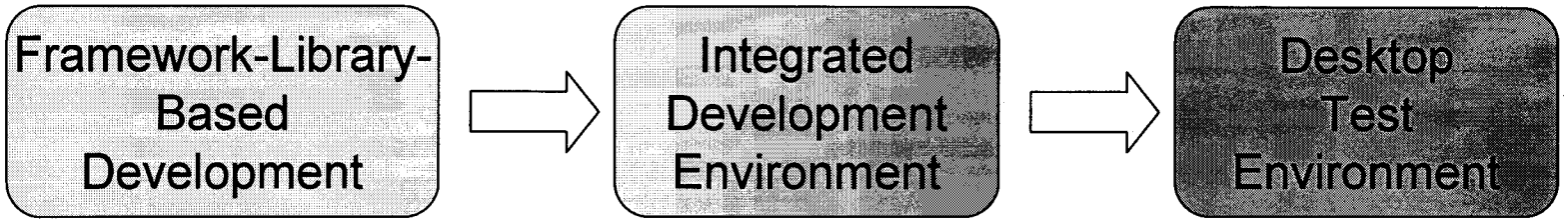


LRO (2009)

Recent GSFC GN&C FSW History

- **MAP**
  - Object-based design (in ANSI C) but not mature enough to support a reuse library
- **ST-5**
  - Successfully tailored MAP design for it's needs
  - Controller class hierarchy useful
- **SDO**
  - Started with MAP's design
  - Shared some code with MAP but no formal reuse
- **GPM** (sent out of house) & **HRV** (cancelled)
  - Advanced MAP's design to be a framework
- **LRO**
  - Using a framework based development process
- **GPM** (back in house)
  - Using a framework-library development process

The Big Picture



Three phased bottom-up approach:

1. Establish framework-library based development environment
2. Create Integrated Development Environment (IDE)
3. Create Non-real-time desktop test environment

Framework-Library Based Development (1 of 2)

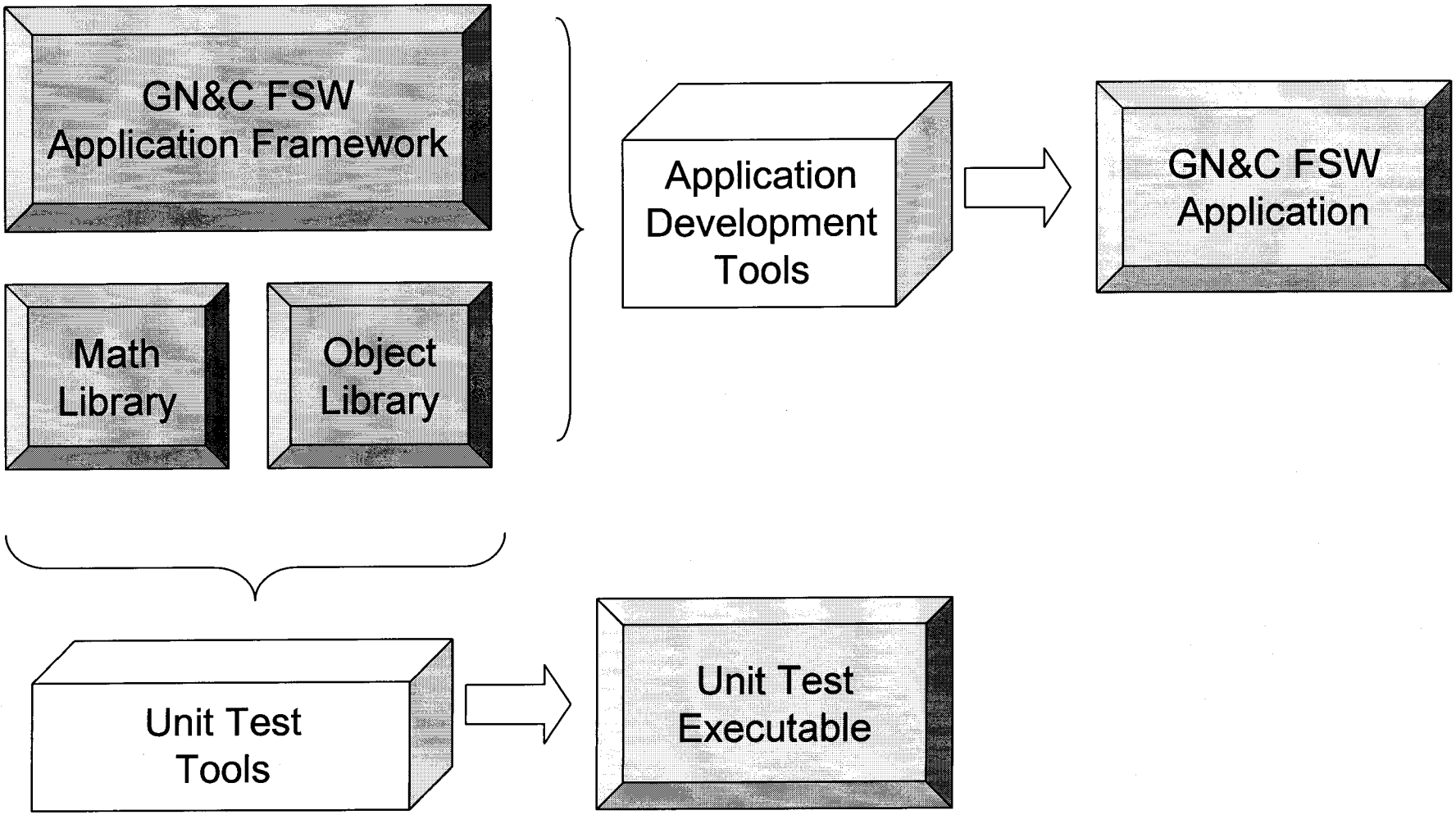
- **Goal**
  - Create a GN&C FSW development infrastructure that supports cost-effective reliable code reuse
- **Develop GN&C *application framework* that supports reusable objects**
  - Framework relies on stable core Flight Executive (cFE) Application Programmer's Interface (API)
  - Framework viable because GN&C applications typically follow a common *pipeline* design
    - Series of objects that produce and consume data
- **A framework is a collection of FSW modules that can be tailored to a meet a mission's specific requirements**
  - Provides explicit tailoring and extension points
- **Define rules and guidelines for using the framework and for developing reusable objects**
  - All reusable FSW must have unit tests and documentation

Framework-Library Based Development (2 of 2)

- **Unit test tools**
  - Assist unit testing
  - Allows consistent library unit tests helping library maintenance

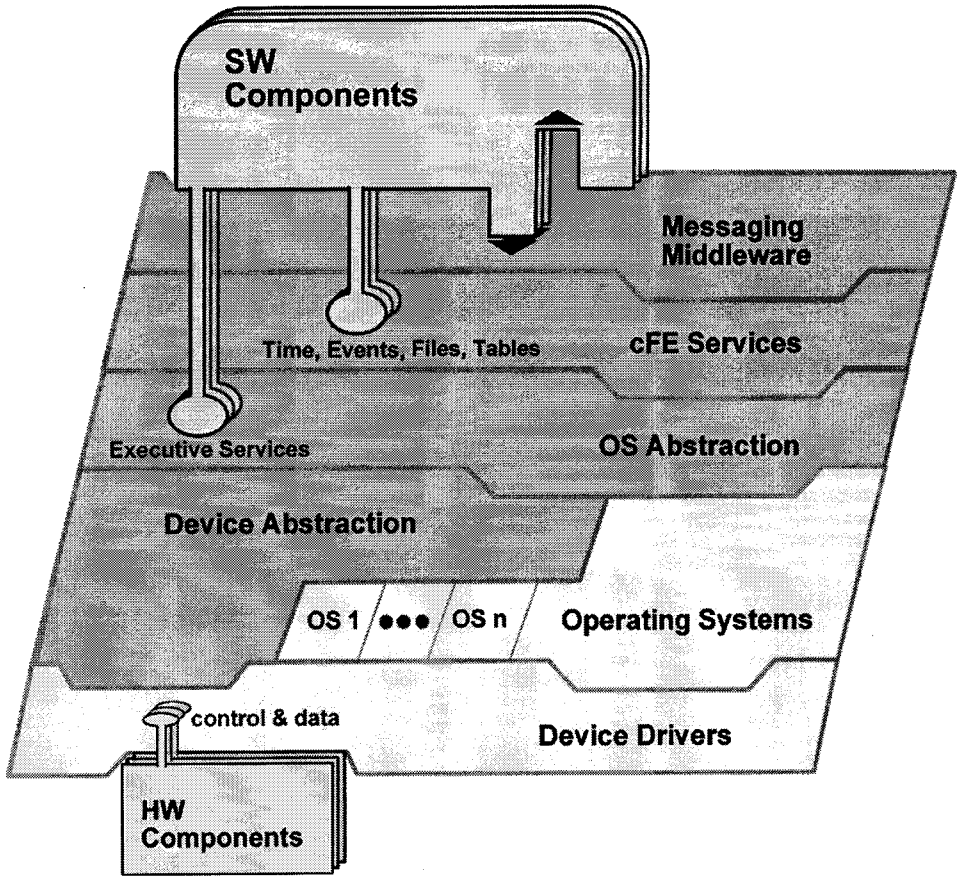
- **Mission cost-benefit**
  - Small learning curve that would almost be negligible if library established as branch standard method of business
  - Consistent application designs
  - Simplified unit testing
  - Current missions will help populate the library with minimal impact
  - Later missions would benefit from existing assets and continue to expand the library

Framework-Library Based Development Processes

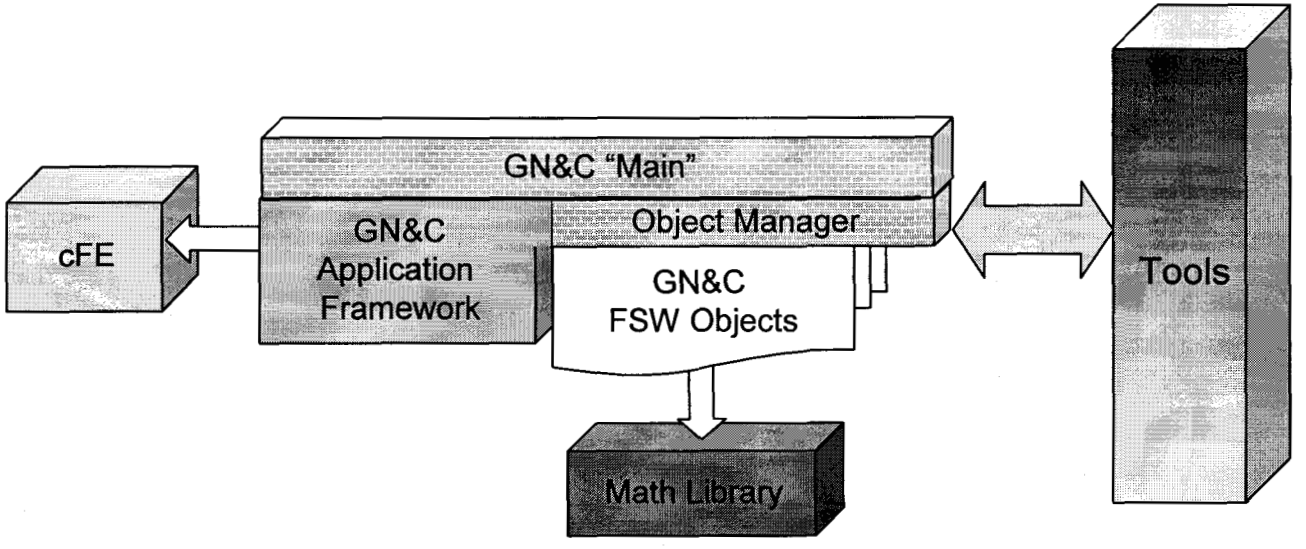


core Flight Executive (cFE) Overview

- Provides common flight executive functions
- Well documented application programmer interface (API)
- Project-independent configuration management
- Applications do not perform any platform specific dependent calls



GN&C FSW Application Framework Architecture



GN&C Application Framework

- Provides standard application infrastructure with an API
- Layered architecture
- Project independent configuration management
- Implemented as a shared library

GN&C FSW Framework Architecture Notes (1 of 2)

- **GN&C Application Framework**
  - Top-level container (App_Frame)
    - Contains and coordinates an application's use of the cFE and GN&C framework utilities
  - cFE Utilities
    - Provide *standard* mechanisms for managing some of the interfaces to the cFE
    - E.g. registering command callback functions
  - GN&C Utilities
    - Provide *standard* mechanisms and infrastructure for creating GN&C FSW Applications
    - Contains an object manager that is used to coordinate the initialization and execution of reusable and mission-specific objects
    - E.g. fault detection reporting utility
- **Templates**
  - Application main and object manager
  - Candidates for automatic code generation

GN&C FSW Framework Architecture Notes (2 of 2)

- **Math Library**
  - Developed by the FSW branch prior to the start of SDO and GPM
  - Based on heritage math libraries
  - Code, unit test, and documentation in branch reuse library

- **GN&C FSW Objects**
  - Provide functionality that meets a mission's functional requirements
  - Can come from the GN&C FSW Library or written specifically for a mission

GN&C FSW Reusable Objects

- **Two classes of objects defined based on the object's dependencies**

- **Framework-independent Objects**
  - Objects that at most depend on branch standard common_types.h and math libraries
  - Easily reusable in other environments
    - Allow analyst to integrate objects into their simulations
  - E.g. solar and lunar models

- **Framework-dependent Objects**
  - Have FSW architectural dependencies beyond framework-independent object dependencies
  - Reusable within the GN&C framework
  - For example the spacecraft ephemeris ground interface object
    - Has cFE event service and GN&C fault detection dependencies

- **Unit test tools organized according to these dependencies**

Conclusion

- **Infrastructure is in place to mature a GN&C FSW framework-based development process**
  - **Mission independent configuration management**
  - **Mature framework design**
  - **Object library design, policy, and procedures under development**
- **LRO**
  - **Using the cFE and the GN&C application framework**
- **GPM**
  - **Mature object library concept**
  - **Start populating the object library**
- **MMS**
  - **Mature requirement process and IDE**